УДК 519.6

N.S. Lesna, T.B. Shatovska, A.V. Lyakhovets

Kharkov national university of radioelectronics, Kharkov

RESEARCH OF DYNAMIC CLUSTERING ON LINEARLY INSEPARABLE PLAGUED EXPERIMENTAL DATA

The system is representing hierarchical multilevel approach of clustering and explores dynamic modeling in hierarchical clustering. Presented approach operates on a k-nn graph and hyper graph in which nodes represent data items, and weighted edges represent similarities among the data items. Presented hierarchical multilevel algorithm consists of several stages: building the graph, coarsening, partitioning, uncoarsening. Exploring of different combinations of algorithms on different stages for different data sets is the main goal of the work.

Keywords: *cluster analysis, hierarchical clustering, coarsening, partitioning, uncoarsening, Chameleon.*

Introduction

The goal of data clustering, also known as cluster analysis, is to discover the natural grouping(s) of a set of patterns, points, or objects [1]. Webster defines cluster analysis as "a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics" [2]. Cluster analysis has been widely used in numerous applications, including pattern recognition, data analysis, image processing and market research [3].

Data clustering is under vigorous development. Contributing areas of research include data mining, statistics, machine learning, spatial database technology, biology and marketing. Owing to the huge amounts of data collected in databases, cluster analysis has recently become a highly active topic in data mining research [4].

The purpose of the work is research of hierarchical multilevel algorithms, investigation of different approaches for different stages of the algorithms, comparison of results of different combinations of the approaches for certain data. Proposed approach is based on the chameleon algorithm. For this research is necessary to collect and analyze methods of graph building, coarsening, partitioning and uncoarsening algorithms.

Basis of the work

Hierarchical clustering is an iterative procedure in which n data points are partitioned into groups which may vary from a single cluster containing all n points, to n clusters each containing a single point.

Presented approach is based on the Chameleon algorithm. Chameleon finds the clusters in the data set by using a two-phase algorithm. During the first phase, Chameleon uses a graph-partitioning algorithm to cluster the data items into several relatively small subclusters. During the second phase, it uses an algorithm to find the genuine clusters by repeatedly combining these subclusters. Chameleon represents its objects based on the commonly used k-nearest neighbor graph approach. This graph representation of the data set allows CHAMELEON to scale to large data sets. Figure 1 provides an overview of the overall approach used by CHAMELEON to find the clusters in a data set [5].

On the basis of the CHAMELEON algorithm presented algorithms will contain 4 main steps:

1. Graph building. Graph will be build by symmetrical k-nn algorithm and asymmetrical k-nn algorithm. For both algorithms different types of distances will be used: Euclidean distance, Manhattan distance, Minkowski distance and Squared Euclidean distance.

2. Graph coarsening. This stage will be implemented by next algorithms: Random Matching, Heavy Edge Matching, Light Edge Matching, Heavy Clique Matching, Heavy-triangle matching, Heaviest Schema Matching.

3. Graph partitioning. This phase will be presented by: Multilevel Recursive Bisection; Kernighan-Lin Algorithm (KL) – Fiduccia-Mattheyses Algorithm; Graph Growing Algorithm (GGP); Greedy Graph Growing Algorithm (GGGP); Multilevel k-way Partitioning; Coordinate Nested Dissection(CND); Recursive Inertial Bisection(RIB); Space-filling Curve Techniques; Sphere-cutting Approach; Levelized Nested Dissection (LND).

4. Graph uncoarsening. This stage will be made by two ways: Kernighan-Lin Refinement - Fiduccia-Mattheyses Algorithm and Boundary Kernighan-Lin Refinement.

Description of the presented system

At different stages of Chameleon algorithm different approaches can be used. For the constructing space graph phase symmetric and asymmetric k-NN algorithms can be used. Different types of distances can be used while graph building. All other phases also can by done by different methods. Such as:

– euclidean distance:

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2};$$

– manhattan distance:

$$d(X_i, X_j) = \sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|;$$

– minkowski distance:

$$d(X_i, X_j) = \left(\sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|^{p} \right)^{1/p};$$

– squared Euclidean distance:

$$d(X_i, X_j) = \sum_{k=1}^{n} (x_{ik} - x_{jk})^2 .$$

The next phase of the algorithm is the coarsening phase. The goal of the coarsening phase is to reduce the size of a graph while preserving those of its properties that are essential to finding a good partition. The original graph is regarded as a weighted graph, with a unit weight assigned to each edge and each node [6]. At the coarsening phase next algorithms can be used.

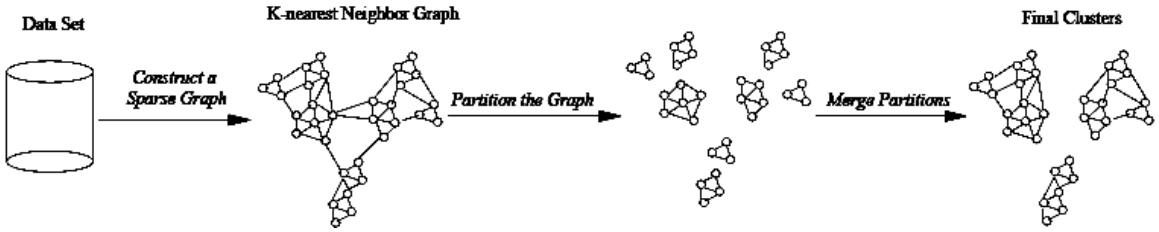

Fig.1. Overall framework CHAMELEON.

Random Matching (RM) A maximal matching can be generated efficiently using a randomized algorithm. The vertices are visited in random order. If a vertex u has not been matched yet, then we randomly select one of its unmatched adjacent vertices. If such a vertex v exists, we include the edge (u; v) in the matching and mark vertices u and v as being matched. If there is no unmatched adjacent vertex v, then vertex u.

Heavy Edge Matching (HEM) Random matching is a simple and efficient method to compute a maximal matching and minimizes the number of coarsening levels in a greedy fashion. However, our overall goal is to find a partition that minimizes the edge-cut. Light Edge Matching (LEM) Instead of minimizing the total edge weight of the coarser graph, one might try to maximize it.

Heavy Clique Matching (HCM) HCM is very similar to the HEM scheme. The only difference is that HEM matches vertices that are only connected with a heavy edge irrespective of the contracted edge-weight of the vertices, whereas HCM matches a pair of vertices if they are both connected using a heavy edge and if each of these two vertices has high contracted edge-weight [7].

Heavy-triangle matching (HTM) coalesces three nodes at a time by picking an unmatched node at random and matching it with two of its neighbours such that the sum of the weights of the three edges connecting the three nodes is maximized over all pairs of neighbours of the selected node. A nonexistent edge between the two neighbours is regarded as an edge of weight zero [8].

Heaviest Schema Matching (HSM). In HSM vertexes are no longer visited in random order. The edges are sorted by their weight and the vertices with the maximum weight edge are selected to do the matching first [6].

The next phase is partitioning. At this phase of a multilevel algorithm computes a high-quality bisection (i.e., small edge-cut) P_m of the coarse graph $G_m = (V_m, E_m)$ such that each part contains roughly half of the vertex weight of the original graph. Since during coarsening, the weights of the vertices and edges of the coarser graph were set to reflect the weights of the vertices and edges of the finer graph, G_m contains sufficient information to intelligently enforce the balanced partition and the small edge-cut requirements. A partition of G_m can be obtained using various algorithms such as [9]:

Multilevel Recursive Bisection;

Kernighan-Lin Algorithm (KL) – Fiduccia-Mattheyses Algorithm;

Graph Growing Algorithm (GGP);

Greedy Graph Growing Algorithm (GGGP);

Multilevel k-way Partitioning;

Coordinate Nested Dissection(CND);

Recursive Inertial Bisection(RIB);

Space-filling Curve Techniques;

Sphere-cutting Approach;

Levelized Nested Dissection (LND).

After the partitioning graph is refining (uncoarsening). During the uncoarsening phase, the partition P_m of the coarser graph G_m is projected back to the original graph, by going through the graphs $G_{m-1}; G_{m-2}; ...; G_1$. Next algorithms can be used for refinement [7]: Kernighan-Lin Refinement – Fiduccia-Mattheyses Algorithm; Boundary Kernighan-Lin Refinement.

Experimental results – Graph building

As implementation of the first phase of the algorithm graph can be presented. Experiments for two data sets will be presented.

At the Figure 2 asymmetrical (k = 4) and symmetrical (k = 7) k-NN graphs built by the different distances for disc-in- disc data set are presented. Data set contains 2000 elements. At the Figure 3 asymmetrical and symmetrical k-NN graphs (k = 3) built by the different distances are for optics-word data set presented. Data set contains 86 elements. Processing time for building graphs are presented in the table 1.

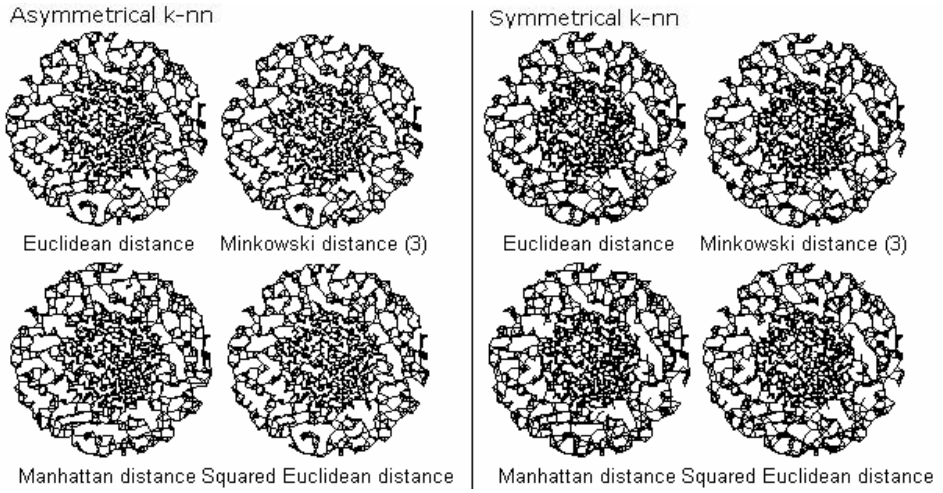

Fig. 2. Asymmetrical k-NN graphs (left), symmetrical k-NN graphs (right) for disc-in-disc data set

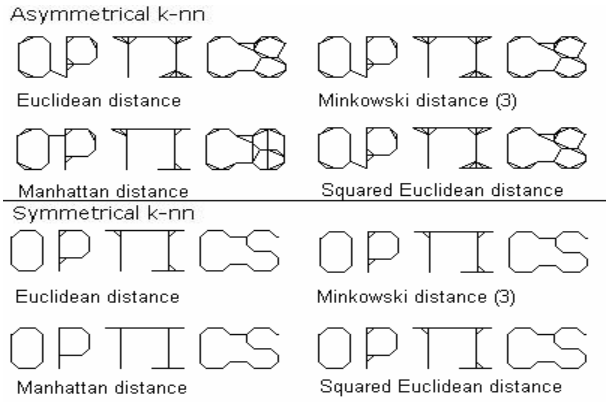

Fig. 3. Asymmetrical k-NN graphs, symmetrical k-NN graphs for "optics" data set

Table 1

Processing time – Buiding the graphs

	Disc-in-disc data set		Optics data set	
	Symmetrical k-nn	Assymetrical k-knn	Symmetrical k-nn	Assymetrical k-knn
Euclidean distance	34.901	4.363	0.0156251	0.0312502
Manhattan distance	34.587	4.667	0.0000000	0.0312502
Minkowski distance(3)	34.68	4.431	0.0000000	0.0312502
Squared Euclidean distance	29.631	4.074	0.0000000	0.0156251

Experimental results – Coarsening

As implementation of the second phase of the algorithm graph can be presented. Experiments for two data sets will be presented. First experiment for optics data set will be presented. At the Figure 4 graphs built by Euclidean distance are presented.

At the Figure 5 graphs built by Manhattan distance are presented. At the Figure 6 graphs built by Minkowski (p = 3) distance are presented. At the Figure 7 graphs built by Squared Euclidean distance are presented. Processing time for coarsening algorithms is presented in table 2.

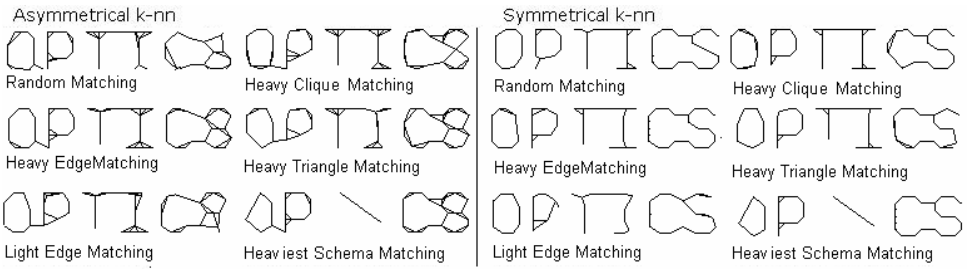

Fig. 4. Euclidean distance

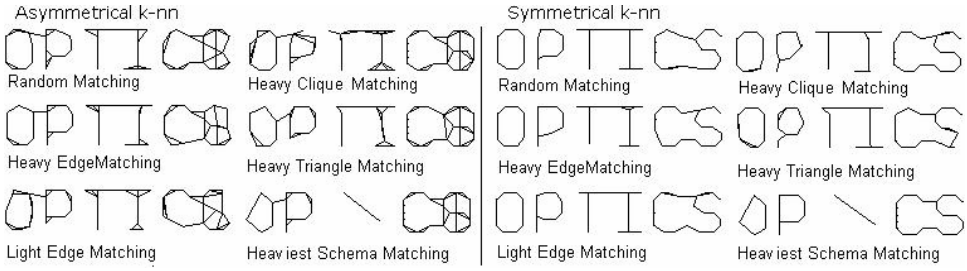

Fig. 5. Manhattan distance

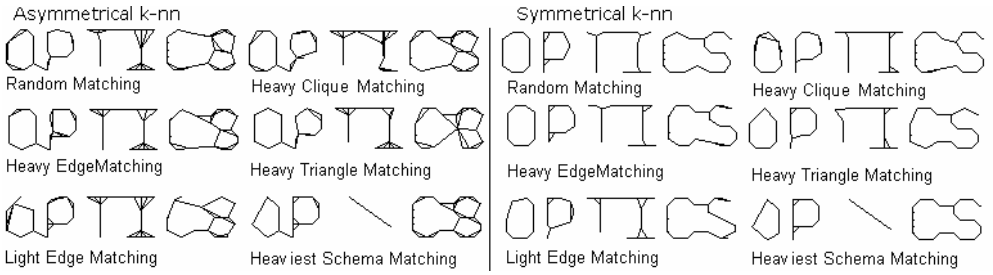

Fig. 6. Minkowski (p = 3) distance

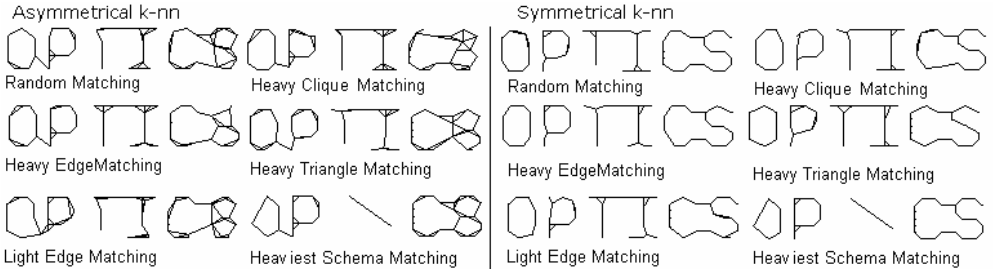

Fig. 7. Squared Euclidean distance

Table 2

Coarsening processing time – optics data set

	Squared Euclidean distance		Minkowski (p=3) distance		Manhattan distance		Euclidian distance	
	Sym	Asym	Sym	Asym	Sym	Asym	Sym	Asym
RM	0,0000000	0,001	0,001	0,0000000	0,0000000	0,001	0,01	0,003
HEM	0,0000000	0,0000000	0,001	0,001	0,0000000	0,001	0,0000000	0,002
LEM	0,0000000	0,001	0,001	0,001	0,001	0,001	0,0000000	0,002
HCM	0,0000000	0,001	0,001	0,001	0,001	0,002	0,0000000	0,003
HTM	0,001	0,001	0,001	0,002	0,001	0,001	0,0000000	0,004
HSM	0,004	0,009	0,005	0,008	0,005	0,008	0,004	0,008

Next the experiment for disc-in-disc data set will be presented.

At the Figure 8 graphs built by Euclidean distance are presented. At the Figure 9 graphs built by Manhattan distance are presented. At the Figure 10 graphs built by Minkowski (p = 3) distance are presented. At the Figure 11 graphs built by Squared Euclidean distance are presented.

Processing time for coarsening algorithms is presented in table 3.

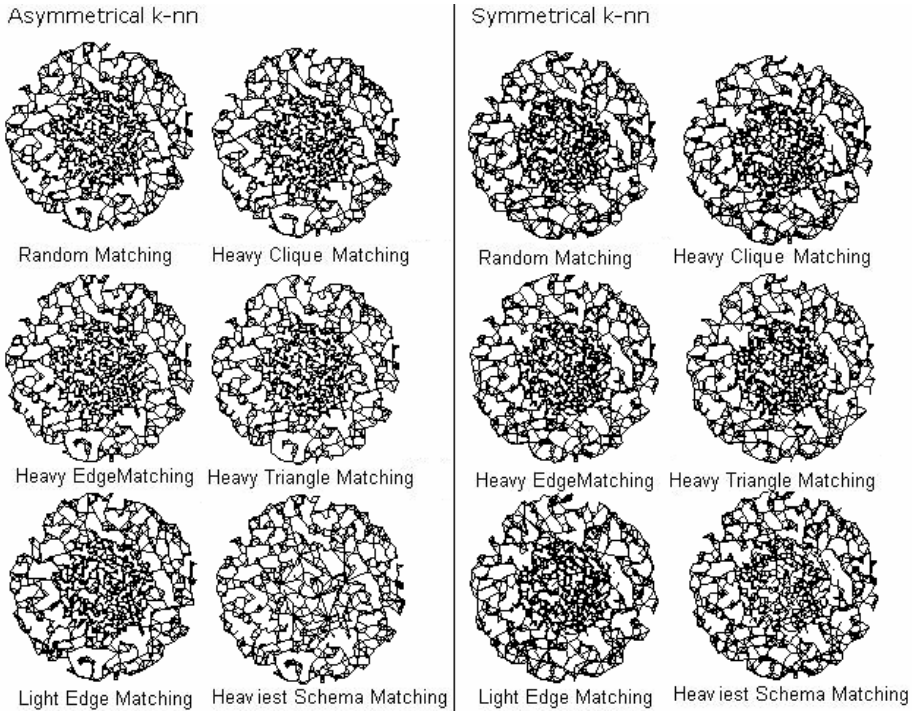

Fig. 8. Euclidean distance

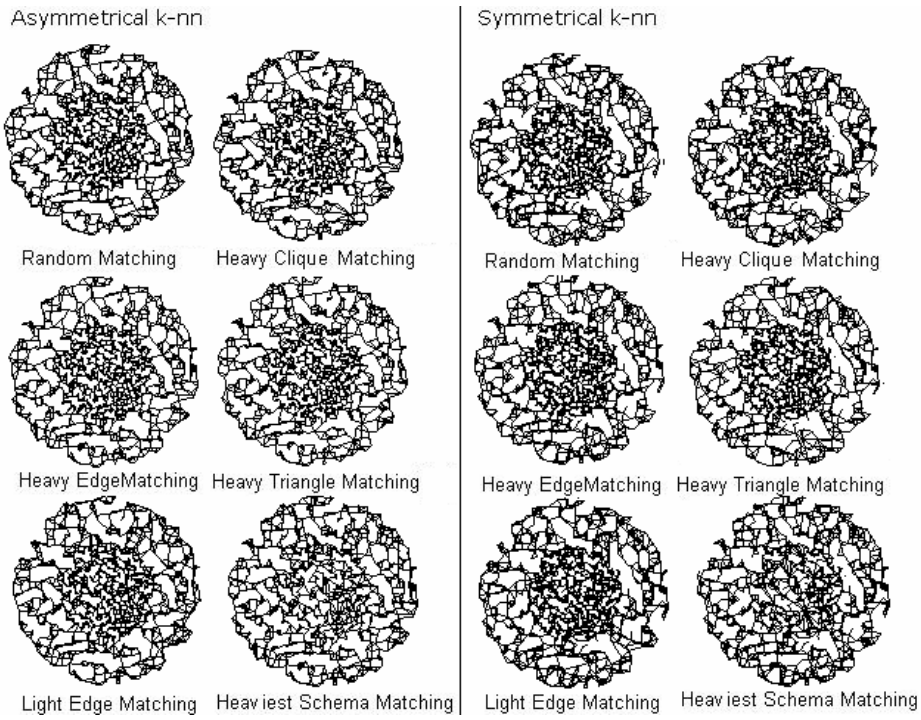

Fig. 9. Manhattan distance

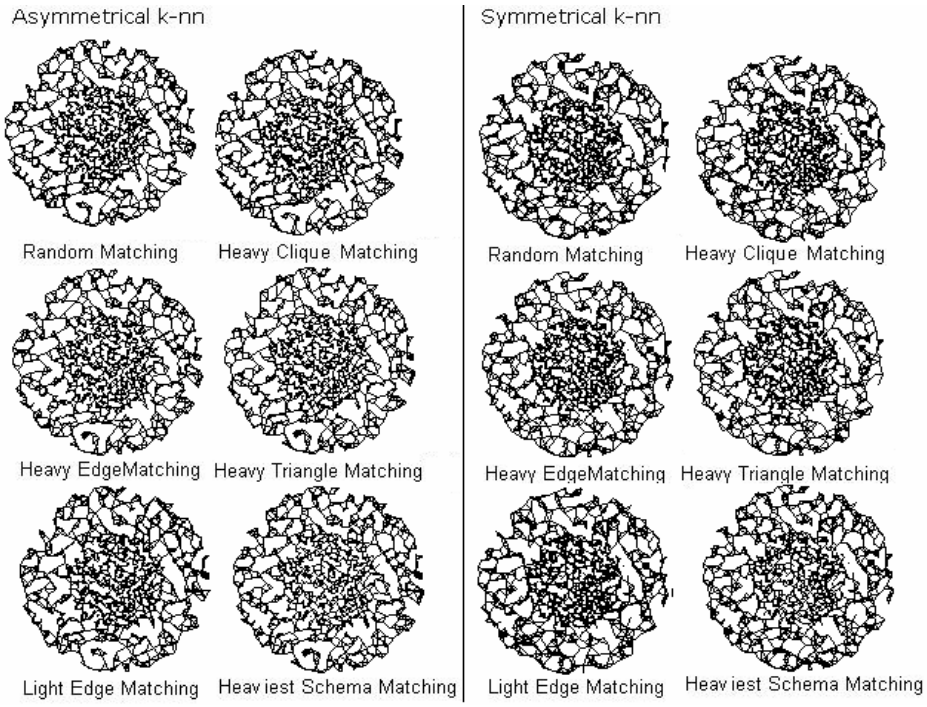

Fig. 10. Minkowski (p = 3) distance

Table 3

Processing time – disc

	Squared Euclidean distance		Minkowski (p=3) distance		Manhattan distance		Euclidian distance	
	Sym	Asym	Sym	Asym	Sym	Asym	Sym	Asym
RM	0.749	0.545	0.781	0.558	0.814	0.565	0.735	0.542
HEM	0.661	0.485	0.65	0.469	0.644	0.531	0.64	0.471
LEM	0.799	0.588	0.792	0.584	0.804	0.603	0.779	0.592
HCM	1.623	1.162	1.579	1.128	1.568	1.156	1.604	1.143
HTM	1.672	1.134	1.685	1.059	1.614	1.123	1.65	1.091
HSM	4:23.915	3:03.808	3:42.401	3:4.689	4:36.834	3:08.674	3:46.671	3.08.096

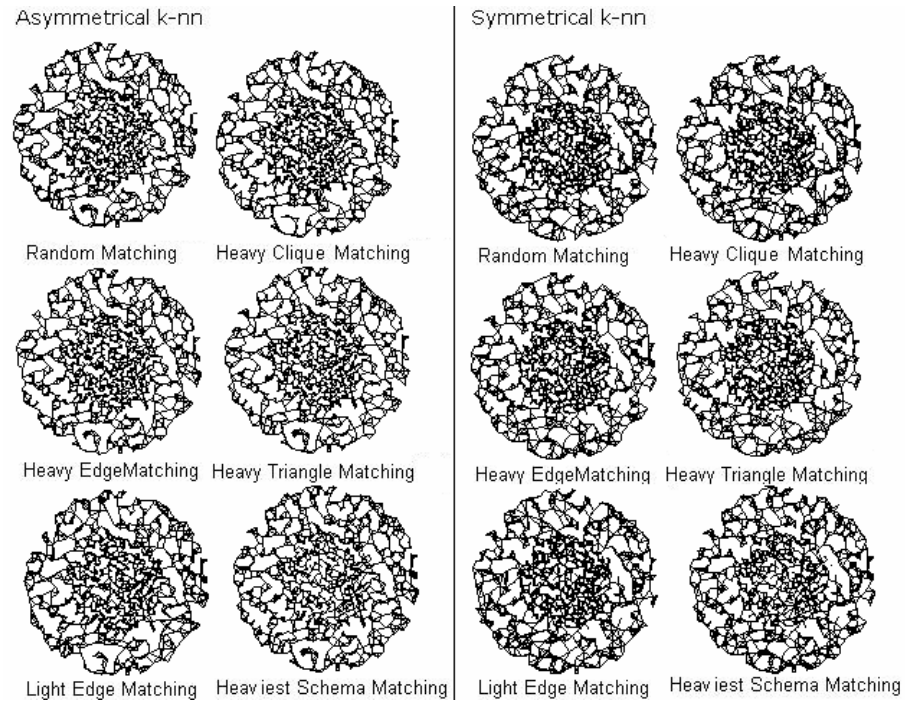

Fig. 11. Squared Euclidean distance

Conclusion and Direction for Future Research

In presented work method for effectiveness research of dynamic clustering of linearly inseparable plagued experimental data is presented. All steps of hierarchical multilevel algorithm are shown up.

Algorithms possible for using on each phase are described. The goal of the whole research is to analyze different combinations of algorithms on different phases. In current work comparison of graph building algorisms and coarsening algorithms were done. In further work comparison of partitioning and uncoarsening algorisms will be done, analysis and estimation of whole schema work will be presented.

References

1. Jain A. Data Clustering: A User's Dilemma / A. Jain, Law M. //Pattern Recognition and Machine Intelligence, First International Conference, PReMI 2005. Proceedings – Kolkata, India, 2005.

2. MERRIAM-WEBSTER ONLINE DICTIONARY. [Web resource] "cluster analysis": http://www.merriam-webster-online.com.

3. Sumathi S. Fundamentals of relational database management systems (Studies in Computational Intelligence) / S. Sumathi, S. Esakkirajan. – NY.: Springer, 2001. – 422 p.

4. Han J. Data Mining: Concepts and Techniques / J. Han, M Kamber. – MORGAN KAUFMANN PUBLISHERS – 2000.

5. Karypis G. CHAMELEON: A hierarchical clustering algorithm using dynamic modelling / G. Karypis, E.H. Han, V. Kumar // In the Proc. IEEE Computer. – 1999. –32 (8). – P. 68-75..

6. Zhu Y. An effective data placement strategy for XML documents / Y. Zhu, K. Lü // Proceedings of the 18th British National Conference on Databases: Advances in Databases. – Chilton, UK, 2001.

7. Karypis G. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs / G. Karypis, V. Kumar // SIAM Journal on Scientific Computing.

8. Gupta A. Fast and efficient algorithms for graph partitioning and sparse matrix ordering / A. Gupta // Computer science. Mathematics. – NY.: Yorkton Heights, 1996.

9. Schloegel K. Graph Partitioning for High Performance Scientific Simulations / K. Schloegel, G. Karypis, V. Kumar // CRPC Parallel Comp. Handbook – Morgan Kaufmann, 2000.

Надійшла до редколегії 19.04.2010

Рецензент: канд. техн. наук, проф. Б.В. Шамша, Харківський національний університет радіоелектроніки, Харків.

ДОСЛІДЖЕННЯ ДИНАМІЧНОЇ КЛАСТЕРИЗАЦІЇ ЛІНІЙНОНЕПОДІЛЬНИХ ЗАШУМЛЕНИХ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ

Н.С. Лесна, Т.Б. Шатовська, А.В. Ляховець

Розглядається ієрархічний багаторівневий алгоритм, що складається з кількох фаз: побудова графу, огрублення, поділ та. об'єднання, на кожній з яких можуть бути використані різні підходи та алгоритми.

Ключові слова: *кластерний аналіз, ієрархічне групування, огрублення, поділ, неогрублення, Chameleon.*

ИССЛЕДОВАНИЕ ДИНАМИЧЕСКОЙ КЛАСТЕРИЗАЦИИ ЛИНЕЙНО НЕРАЗДЕЛИМЫХ ЗАШУМЛЕННЫХ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Н.С. Лесная, Т.Б. Шатовская, А.В. Ляховец

Рассматривается иерархический многоуровневый алгоритм, состоящий из нестольких фаз: построение графа, огрубление, разделение и восстановление, на каждой из которых могут быть использованны различные подходы и алгоритмы.

Ключевые слова: *кластерный анализ, иерархическое группирование, огрубление, разделение, неогрубление, Chameleon.*